Michael A. Burke, Esq. (NSB #11527)
ROBISON, SHARP, SULLIVAN & BRUST
71 Washington St.
Reno, NV 89503
Telephone: (775) 329-3151
Email: mburke@rssblaw.com

Andrea Pacelli, Esq. (*pro hac vice*)
Mark S. Raskin, Esq. (*pro hac vice*)
Michael DeVincenzo, Esq. (*pro hac vice*)
Elizabeth Long, Esq. (*pro hac vice*)
Eric Berger, Esq. (*pro hac vice*)
KING & WOOD MALLESONS LLP
500 Fifth Ave., 50th Floor
New York, New York 10110
Telephone: (212) 319-4755
Email: andrea.pacelli@us.kwm.com
mark.raskin@us.kwm.com
michael.devincenzo@us.kwm.com
elizabeth.long@us.kwm.com
eric.berger@us.kwm.com

Steven C. Sereboff, Esq. (*pro hac vice*)
SoCAL IP LAW GROUP LLP
1332 Anacapa, Suite 201
Santa Barbara, CA 93101
Telephone: (805) 230-1356
Email: ssereboff@socalip.com

Attorneys for Plaintiff
Applications in Internet Time LLC

UNITED STATES DISTRICT COURT

DISTRICT OF NEVADA

APPLICATIONS IN INTERNET TIME, LLC, Plaintiff, v. SALESFORCE.COM, INC., Defendant.	Civil Action No.: 3:13-CV-00628-RCJ-CLB **PLAINTIFF APPLICATIONS IN INTERNET TIME, LLC'S OPENING CLAIM CONSTRUCTION BRIEF**

TABLE OF CONTENTS

TABLE OF AUTHORITIES

I. INTRODUCTION

The words of a claim define the metes and bounds of a patented invention. The claim construction process exists to provide guidance to the finder of fact with respect to the meaning of the words used in the claim. To the extent this Court elects to construe the disputed claim terms, Applications in Internet Time, LLC's ("AIT") proposed constructions are correct because they conform to the ordinary meaning of the claim language and are consistent with both the specification and the prosecution history of United States Patent Nos. 7,356,482 (the "'482 Patent," Ex. A[1]) and 8,484,111 (the "'111 Patent," Ex. B) (collectively, the "Asserted Patents").

In contrast to AIT's proposed constructions, salesforce.com, inc.'s ("Salesforce") proposed constructions do not actually define the words used in the claims. Instead, Salesforce's "constructions" seek to read into the claims extraneous requirements not found on the face of the claim language in a transparent attempt to manufacture defenses where none exist. Salesforce's constructions fly in the face of established Federal Circuit precedent which repeatedly denounces the tactic employed by Salesforce as the "cardinal sin" of claim construction.

II. BACKGROUND

A. Integrated Change Management Unit Technology

In the late 1990s and early 2000s, Alternative Systems, Inc. ("ASI") developed and commercialized software for use by various third-party businesses in the environmental, health and safety fields. In the late 1990s, software developers, like ASI, servicing third party businesses needed to constantly update their software to reflect varying customer needs. To address that problem ASI developed and commercialized a new software technology, called Integrated Change Management Unit ("ICMU"), that allowed the user interface and functionality of an application to be modified without having to reprogram the underlying code. On December 18, 1998 ASI filed the patent application that led to the Asserted Patents based on that technology. In 2012, ASI assigned the Asserted Patents to AIT, a company led by Douglas Sturgeon, a creator of the ICMU technology and named inventor on the patents.

[1] Unless indicated otherwise, all exhibits refer to the documents attached to the Declaration of Andrea Pacelli in Support of AIT's Opening Claim Construction Brief filed herewith.

The Asserted Patents are directed to a software architecture (the ICMU technology) that allows applications to be designed and maintained without requiring the reprogramming of the underlying software code. This feature addressed the problem of "continual reprogramming of the database software" required for the maintenance of software applications. Ex. A, '482 Patent, at 8:5.[2] One of the features that enables the ICMU architecture to overcome that problem is its use of a "data-driven" architecture. *Id.* at 10:17–20 ("Unlike 'hard-coded' systems, in which business functionality and content is managed by explicit lines of code, the metadata architecture of the invention is entirely data-driven."). The use of the data-driven architecture not only avoided the need for reprogramming; it also allowed application development to be accomplished more simply, such as by non-programmers. The Asserted Patents explain that the "[n]ormal programming steps are decomposed into pieces that can be combined by a non-programmer into a coherent set of procedures that define a unique system." *Id.* at 15:46–49.

B. The Asserted Patents

Claim 1 of the '482 Patent is exemplary and reads:

> 1. A system for providing a dynamically generated application having one or more functions and one or more user interface elements; comprising:
>
> > a server computer;
> >
> > one or more client computers connected to the server computer over a computer network;
> >
> > a first layer associated with the server computer containing information about the unique aspects of a particular application;
> >
> > a second layer associated with the server computer containing information about the user interface and functions common to a variety of applications, a particular application being generated based on the data in both the first and second layers;
> >
> > a third layer associated with the server computer that retrieves the data in the first and second layers in order to generate the functionality and user interface elements of the application; and
> >
> > a change management layer for automatically detecting changes that affect an application,
> >
> > each client computer further comprising a browser application being executed

[2] The Asserted Patents have substantially identical specifications. To avoid confusion all citations to the specification refer to U.S. Patent No. 7,356,482.

> by each client computer, wherein a user interface and functionality for the particular application is distributed to the browser application and dynamically generated when the client computer connects to the server computer.

Ex. A, '482 Patent, at 32:9–34. The claim recites a "server computer" which supports a stack of four software "layers," each layer providing a specific functionality. As depicted in Fig. 1, in a disclosed embodiment each of the four layers (indicated as 17, 15, 13, and 11, respectively) communicates with the layers above and below it:

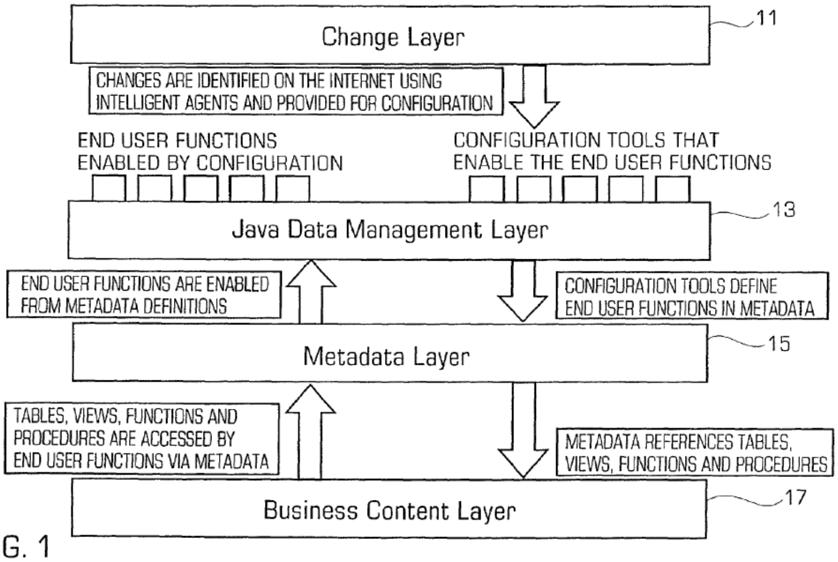
Change Layer
11
CHANGES ARE IDENTIFIED ON THE INTERNET USING INTELLIGENT AGENTS AND PROVIDED FOR CONFIGURATION
END USER FUNCTIONS ENABLED BY CONFIGURATION
CONFIGURATION TOOLS THAT ENABLE THE END USER FUNCTIONS
Java Data Management Layer
13
END USER FUNCTIONS ARE ENABLED FROM METADATA DEFINITIONS
CONFIGURATION TOOLS DEFINE END USER FUNCTIONS IN METADATA
Metadata Layer
15
TABLES, VIEWS, FUNCTIONS AND PROCEDURES ARE ACCESSED BY END USER FUNCTIONS VIA METADATA
METADATA REFERENCES TABLES, VIEWS, FUNCTIONS AND PROCEDURES
Business Content Layer
17

FIG. 1

Each of the layers is described below.

Business Content Layer. Starting from the bottom, the business content layer 17 contains business content including information regarding the unique aspects of a particular application. The specification explains that "[w]ithin the business content layer, the relevant items are stored (and changed, as appropriate) for the specific business operations of concern to the end user." *Id.* at 9:56–59. The specification further explains that the "business content" contained in this layer may "include[] business knowledge, logical designs, physical designs, physical structures, relationships, and data associated with a selected area of business activity." *Id.* at 12:16–20.

Metadata Layer. The next component in the embodiment of Fig. 1 is the metadata layer 15, which contains information about the user interface and functions common to a variety of applications. *Id.* at 9:41–46 ("[The] metadata layer 15 … provides and/or defines data about every feature of the user interface including, without limitation, tools, worklists, data entry forms, reports, documents, processes, formulas, images, tables, views, columns, and other structures and functions[.]"). The metadata architecture "stores all of the information used to create the front-end business application and manage the back-end business database." *Id.* at 10:15–17.

Java Data Management Layer. The next layer, or Java[3] data management layer, uses the information stored in the business content layer and the metadata layer to generate the functionality and user interface elements of the application. For example, this layer may "provide[] a graphical user interface … which allows a web browser user to communicate with the metadata and business content layers on a server… ." *Id.* at 15:5–10. As the application information is contained in the business content layer and the metadata layer, the Java data management layer is not required to be reprogrammed in response to changes in application requirements. *Id.* at 15:11–16.

Change Management Layer. The topmost layer, or change management layer 11, automatically detects the changes that affect an application. The Asserted Patents recognize that the changes detected by the change management layer may include changes in any factor "that materially affects operations and/or information management requirements of a particular business." *Id.* at 9:2–4. The Asserted Patents teach that the change management layer can detect such changes from local sources, such as an intranet, or external sources, such as a website on the Internet. *Id.* at 16:18–19 ("[t]he change layer primarily involves an intranet or the Internet"); 19:67 (explaining "internal and external Web activities" can be searched for changes). In a disclosed embodiment, the "change management layer 11 … includes one or more change agents that … identify and bring to the user's attention relevant regulatory and non-regulatory changes … that may affect a user's business[.]" *Id.* at 9:34–38.

As depicted by the arrows in Fig. 1, each layer "uses" the layers below it and is "used by" the layers above it. For example, the metadata layer acts as an interface between the business content layer and the Java data management layer. *Id.* at 9:52–56 ("Within the metadata layer, the relevant

[3] Java is a programming language for Internet applications. Java "applets" can be dynamically generated and distributed over the Internet to be executed remotely on a user's browser. *See* Ex. A, '482 Patent, at 14:22–15:4.

items (data entry forms, etc.) in the business content layer are defined, regulatory and non-regulatory changes in these items are implemented, and access thereto is provided."); 9:59–61 ("A business area … in the business content layer is referenced and described by the metadata layer to enable management by the data management layer."). The business content layer "is defined by and referenced in the metadata layer so that the necessary … procedures and data can be read and updated by the Java data management layer." *Id.* at 12:24–28; *see also id.* at 15:17–30 (metadata provides "data mapping" between business content layer and Java data management layer). Finally, the Java data management layer includes "change configuration functions" that enable the change management layer to modify the content of the business content layer and the metadata layer in response to automatic detection of changes that affect an application: "The Change Configuration functions support creation and change of End User functions through a variety of flexible and intelligent manual routines, such as intelligent agents, screens, fields, reports, documents and logic that can be changed without requiring programming skills." *Id.* at 10:6–10.

III. LEGAL STANDARD

A. Claim Construction

As a general proposition, claim construction begins with the words of the claims themselves. *Amgen Inc. v. Hoechst Marion Roussel, Inc.*, 457 F.3d 1293, 1301 (Fed. Cir. 2006). Claim terms are generally given their ordinary and customary meaning, which "is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005). "[T]he claims themselves provide substantial guidance as to the meaning of particular claim terms." *Id.* at 1314. For example, "the context in which a term is used in the asserted claim can be highly instructive," and "[o]ther claims of the patent in question, both asserted and unasserted, can also be valuable sources of enlightenment as to the meaning of a claim term." *Id.* "In some cases, the ordinary meaning of claim language … may be readily apparent even to lay judges, and claim construction in such cases involves little more than the application of the widely accepted meaning of commonly understood words." *Id.*

The specification "is the single best guide to the meaning of a disputed term." *Phillips*, 415 F.3d at 1315. However, care must be taken to avoid reading a requirement into the claims that is not apparent from the ordinary meaning of a term. The Federal Circuit has stated in no uncertain terms that there is a "stringent standard for narrowing a claim term beyond its plain and ordinary meaning," *Aventis Pharma S.A. v. Hospira, Inc.*, 675 F.3d 1324, 1330 (Fed. Cir. 2012). Such a narrowing interpretation is only appropriate under two circumstances: "1) when a patentee sets out a definition and acts as [its] own lexicographer, or 2) when the patentee disavows the full scope of a claim term either in the specification or during prosecution." *Id.* (quoting *Thorner v. Sony Computer Entertainment America L.L.C.,* 669 F.3d 1362, 1365 (Fed. Cir. 2012)). Such special definitions and/or disclaimers must be express and unambiguous. *Arlington Indus., Inc. v. Bridgeport Fittings, Inc.*, 632 F.3d 1246, 1254 (Fed. Cir. 2011) ("[E]ven where a patent describes only a single embodiment, claims will not be read restrictively unless the patentee has demonstrated a clear intention to limit the claim scope using words [or] expressions of manifest exclusion or restriction.").

The importation of requirements from the disclosed embodiments into the claims is the "cardinal sin" of claim construction. *SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys. Inc.*, 242 F.3d 1337, 1340 (Fed. Cir. 2001). That is because the specification itself does not delimit the right to exclude. That is the function and purpose of the claims. *Kara Tech. Inc. v. Stamps.com Inc.*, 582 F.3d 1341, 1348 (Fed. Cir. 2009) ("The claims, not specification embodiments, define the scope of patent protection."); *see Interactive Gift Exp., Inc. v. Compuserve Inc.*, 256 F.3d 1323, 1331 (Fed. Cir. 2001) ("In construing claims, the analytical focus must begin and remain centered on the language of the claims themselves ….").

The next source of potentially relevant information is the prosecution history of a patent. The Federal Circuit has cautioned against placing too much emphasis on the prosecution history because statements in the prosecution history are often not models of clarity. *Phillips*, 415 F.3d at 1317 ("[B]ecause the prosecution history represents an ongoing negotiation between the PTO and the applicant, rather than the final product of that negotiation, it often lacks the clarity of the specification and thus is less useful for claim construction purposes."). As such, the standard for importing requirements from prosecution history statements into the claims is high, demanding "clear and unmistakable" disavowal of claim scope. *3M Innovative Properties Co. v. Tredegar Corp.*, 725 F.3d 1315, 1325 (Fed. Cir. 2013) ("[I]n order for prosecution disclaimer to attach, the disavowal must be both clear and unmistakable."). The Federal Circuit has explained that a "clear and unmistakable disclaimer" cannot exist if "a prosecution argument is subject to more than one reasonable interpretation." *01 Communique Lab., Inc. v. LogMeIn, Inc.*, 687 F.3d 1292, 1297 (Fed. Cir. 2012) (citations and quotations omitted).

Extrinsic evidence "consists of all evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises." *Phillips*, 415 F.3d at 1317. Extrinsic evidence is generally viewed as less reliable than intrinsic evidence and cannot be used to alter the meaning of a claim term based on the claims themselves, the specification, or the prosecution history. *See id.* at 1318–19.

B. Indefiniteness

A party asserting that a patent claim is invalid for indefiniteness bears the burden of proving indefiniteness by clear and convincing evidence. *See Young v. Lumenis, Inc.*, 492 F.3d 1336, 1344–45 (Fed. Cir. 2007). In order to prove indefiniteness, Salesforce must establish that the "claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention." *Nautilus, Inc. v. Biosig Instruments, Inc.*, 572 U.S. 898, 901 (2014). Satisfaction of § 112, ¶ 2 requires only reasonable certainty, not absolute certainty or precision. *Id.* at 910. The Federal Circuit has explained that constructions rendering claims invalid or meaningless should be avoided unless such construction "is the '*only* claim construction that is consistent with the claim's language and the written description.'" *Marine Polymer Techs., Inc. v. HemCon, Inc.*, 672 F.3d 1350, 1368 (Fed. Cir. 2012) (quoting *Rhine v. Casio, Inc.*, 183 F.3d 1342, 1345 (Fed. Cir. 1999)).

IV. ARGUMENT[4]

A. "automatically detect[ing]"

Claim Term/Phrase	AIT Construction	Salesforce Construction
"automatically detect[ing]" ('482 claims 1, 21)	"detect[ing] without direct human intervention"	Indefinite, or in the alternative, requiring at least "detect[ing] without any intervention by a human operator

[4] As part of the claim construction process, the parties have agreed to constructions for a number of claim terms. *See* Ex. C, Agreed Constructions. AIT respectfully requests the Court to adopt those agreed constructions.

		through the use of one or more intelligent agents"

The parties have three disputes with respect to the phrase "automatically detect[ing]." First, the parties dispute what it means to "automatically detect" something. For example, while the parties agree that the step of detecting must be performed without human interaction, AIT's construction clarifies that the human interaction may indirectly be required for certain limited purposes such as initiating the software processes that perform the automatic detection. Second, the parties dispute whether "the use of one or more intelligent agents" should be read into the claims from an embodiment discussed in the specification. Lastly, Salesforce wrongly argues that the term "automatically detect[ing]" is indefinite.

1. Detecting Automatically Does Not Preclude Indirect Human Interactions

With respect to the first issue, there is no dispute that the step of automatically detecting changes must be performed without direct human interaction. However, as with any software process, a certain level of human interaction is required to, *e.g.*, to turn on a device and/or initiate the software process. *See* Ex. H, Rosenberg Reply CC Decl., ¶ 27–28. As such, AIT's construction makes clear for the jury that some level of human interaction is permitted as long as the specific claimed action of detecting changes is not performed by a human. In particular, human interaction may be used to perform indirect steps such as those that occur in order to initiate the detection step. The claim language, the specification, and the prosecution history all support the conclusion that the indirect steps related to "automatically detect[ing]" can involve human interaction. Salesforce's construction is silent on this issue.

The claim language supports AIT's construction. With respect to the term "automatically detect[ing]," the claim language specifies the extent to which automation is required. For example, claim 1 of the '482 Patent recites "[a] system for providing a dynamically generated application … comprising" eight limitations. Ex. A, '482 Patent, at 32:9–34. Yet, only one limitation explicitly requires automation: "a change management layer for automatically detecting changes that affect an application." On its face, the claim language is not directed to a completely automated process with no human intervention. Instead, pursuant to the claim language, the step of detecting changes must be performed without human interaction, but the claimed and unclaimed steps or functions both leading to and resulting from such automated detection are not so limited. As the "automatically detect[ing]" limitation appears as part of a "comprising" claim, the claim language itself permits the use of manual steps as long as they are not directly involved in the detecting step. *See CollegeNet, Inc. v. ApplyYourself, Inc.*, 418 F.3d 1225, 1235 (Fed. Cir. 2005) (construing term "automatically" in software limitation such that a "human may initiate or interrupt the process" based, in part, on use of "comprising"); *Vehicle IP, LLC v. Werner Enterprises, Inc.*, 4 F. Supp. 3d 648, 660 (D. Del. 2013) ("The use of the word 'comprising' … suggests that the asserted limitations should be inclusive of the possibility that manual operations could occur before, after, and/or during the claimed automatic functions.").

The specification supports AIT's construction. AIT's proposed construction is consistent with the use of the term "automatically detect[ing]" in the Asserted Patents. The specification expressly states that a user can choose or elect to either "automatically configure" or "manually implement" the detected change information. Ex. A, '482 Patent, at 10:57–60 ("Configuration Users can choose to automatically configure the preceding recommendation based on a set of default conditions, or can manually implement the configuration using a configuration toolkit."). Fig. 2, reproduced here, shows how a user can control automatic processes to, *e.g.*, accept recommended changes (which may be automatically detected) and choose to automatically configure such changes. As such, the specification teaches that human intervention may be used indirectly to control an automated function or step.

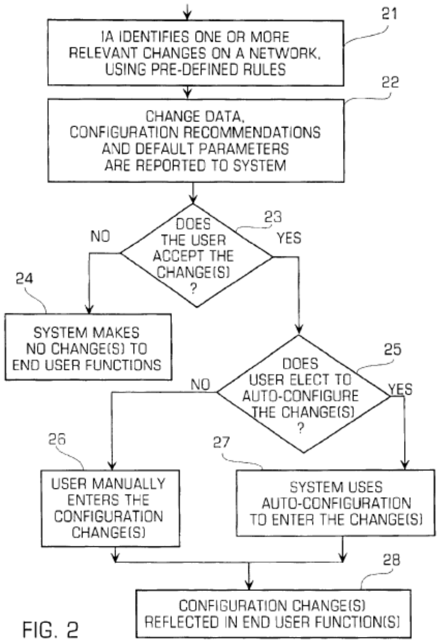
21
IA IDENTIFIES ONE OR MORE RELEVANT CHANGES ON A NETWORK, USING PRE-DEFINED RULES
22
CHANGE DATA, CONFIGURATION RECOMMENDATIONS AND DEFAULT PARAMETERS ARE REPORTED TO SYSTEM
23
DOES THE USER ACCEPT THE CHANGE(S) ?
NO
YES
24
SYSTEM MAKES NO CHANGE(S) TO END USER FUNCTIONS
25
DOES USER ELECT TO AUTO-CONFIGURE THE CHANGE(S) ?
NO
YES
26
USER MANUALLY ENTERS THE CONFIGURATION CHANGE(S)
27
SYSTEM USES AUTO-CONFIGURATION TO ENTER THE CHANGE(S)
28
CONFIGURATION CHANGE(S) REFLECTED IN END USER FUNCTION(S)

FIG. 2

Similarly, the specification teaches that the triggering of an alert message may be performed "automatically" but that a user may be involved indirectly to set up this automated response:

> A user selects an alert mode, specifies an expiration date and time, selects one or more recipients, selects manual/send for the alert, views an alert history and/or sets

up one or more alert conditions that will automatically trigger the sending of an alert. *Id.* at 20:15–20. The specification recognizes that the mere fact that a user manually "sets up one or more alert conditions" does not mean that actions taken based upon or using those alert conditions, such as the sending of an alert, are not performed automatically. *Id.*

The specification further discusses an advantage of the invention as "recommending modifications to the business content, and automatically effecting modifications in the system without the use of programmers and/or programming." *Id.* at 7:64–67. The "recommend[ed] modifications" that are "automatically effect[uated]" require human approval before they are effectuated "automatically." As such, the specification recognizes, the changes that are ultimately detected automatically may be caused by human activity.

In sum, the specification consistently and repeatedly uses the term "automatically" in a manner that permits indirect human activities. As such, any construction of "automatically" that precludes indirect human interaction would be inconsistent with the specification's use of the word "automatically" and should not be adopted. *Phillips*, 415 F.3d at 1322–23 (cautioning that constructions cannot "contradict any definition … ascertained by a reading of the patent documents"). Indeed, as detailed, reading "automatically" to preclude indirect human interaction would exclude a preferred embodiment. Such constructions are "rarely, if ever, correct." *SynQor, Inc. v. Artesyn Techs., Inc.*, 709 F.3d 1365, 1378–79 (Fed. Cir. 2013) ("A claim construction that 'excludes the preferred embodiment is rarely, if ever, correct and would require highly persuasive evidentiary support.'").

The prosecution history supports AIT's construction. Consistent with the claims and the specification, the prosecution history evidences that the term "automatically" does not preclude indirect human interaction. During the prosecution of the application that led to the issuance of the '482 Patent, the examiner rejected the claims as allegedly anticipated by U.S. Patent No. 5,960,200 ("Eager"). Ex. D, 02/28/07 Office Action, at 3. With respect to the "automatically detecting" limitation, the examiner found that the "change management layer for automatically detecting changes" was disclosed in two passages and certain figures describing Eager's "re-architecting system." *Id.* at 4 (relying on "column 2, lines 34-57 and column 4, lines 42-51 and Figs. 1, 16, 17, 24" and finding "specifically, 'change management layer' corresponds to re-architecting system in Eager's teaching"). In response to the examiner's rejection, the inventors appealed the decision to the United States Patent and Trademark Office appeal board.[5]

In their appeal, the inventors explained that both of the passages relied on by the examiner failed to disclose detecting changes in an application, let alone automatically detecting such changes. Ex. E, 08/28/07 Appeal Brief, at 10–12. The inventors conceded that Eager disclosed an automated function, namely "automatically translating" a source application, but explained that functionality "[did] not relate to detecting changes in the applications, and relate[d] even less to automatically detecting such changes." *Id.* at 10–11. The inventors further explained why each of the figures cited by the examiner also failed to disclose "detecting changes that affect an application." *Id.* at 12–13. Plainly, the inventors did not disavow the meaning of "automatically" reflected by the specification in distinguishing the portions of Eager relied on by the examiner. Instead, the inventors' arguments were addressed to the word "detecting" and had nothing to do with the word "automatically."

Separately from addressing Eager's "re-***architecting*** system", the inventors explained that Eager's "re-***engineering*** system" (which the examiner had not relied on) taught away from "automatically detecting changes." The inventors explained that "Eager teaches modifying applications in relation to Eager's re-engineering system." *Id.* at 14. The inventors further explained that Eager stated that these modifications were performed by "application developers and maintenance personnel [who] modify application screens and messages." *Id.* The inventors informed the examiner that this manual modification taught away from a "means that would operate 'automatically.'" *Id.* Far from representing a clear and unmistakable disavowal, the inventors' statement stands for the unremarkable (and undisputed) position that when an act is performed by human developers or maintenance personnel, ***that act*** is not performed automatically.

In sum, the appeal brief: (i) argued that the portions of Eager relied on by the examiner failed to disclose detecting application changes at all, let alone detecting such changes automatically; and

[5] The Board never considered the inventors' arguments, because in response to the appeal brief the examiner withdrew the rejection and allowed the claims. *See* Ex. O, 12/28/07 Notice of Allowability.

(ii) explained that Eager's teaching with respect to "modifying applications screens," which the examiner had not alleged to be detecting changes, was manually performed by "application developers," teaching away from an automated function. The use of the term "automatically" in the appeal brief is entirely consistent with the ordinary meaning of that term reflected by the specification. There are no statements remotely directed to, let alone expressly and unequivocally disclaiming, indirect actions by a user with respect to the claimed "automatically" detecting steps. *3M Innovative*, 725 F.3d at 1325 ("[I]n order for prosecution disclaimer to attach, the disavowal must be both clear and unmistakable."); *SRI Int'l, Inc. v. Cisco Sys., Inc.*, 930 F.3d 1295 (Fed. Cir. 2019) (in order to read requirements into the claims from the prosecution history, "the prosecution history must evince a 'clear and unmistakable surrender'" of claim scope).

Precedent confirms that the term "automatically" does not preclude indirect human activities. Where, as here, the claim language only requires a single software step or function to be performed "automatically," it is legally incorrect to construe the word "automatically" to preclude all human intervention at any level.

The Federal Circuit addressed a similar issue to the one presented here in *CollegeNet, Inc. v. ApplyYourself, Inc.*, 418 F.3d 1225 (Fed. Cir. 2005). In that case, the Federal Circuit held that "automatically" means "*once initiated*, the function is performed by a machine, without the need for manually performing the function." *Id.* at 1235–36. The *CollegeNet* court explained that a "machine still performs the claimed functions without manual operation, even though a human may initiate or interrupt the process." *Id.* at 1235. The Federal Circuit provided several examples supporting an understanding of automatic processes that contemplate indirect human intervention, such as an automatic dishwasher and auto-pilot:

> [S]imply because a human has to load [an automatic dishwasher] and press the start button, and has the ability to turn it off mid-cycle, does not mean that the device does not 'automatically' wash the dishes. Similarly, an autopilot which is turned on by a human and necessarily must be able to be interrupted by a human once the automatic process is engaged … remains an 'automatic' device.

Id. (internal citations and quotations omitted). The Federal Circuit further explained that an inclusive definition of "automatically" was proper because the claims in *CollegeNet*, like the claims at issue here, contained the word "comprising." *Id.* ("[T]he use of 'comprising' suggests that additional, unrecited elements are not excluded. Such elements could include human actions to expressly initiate the automatic storing or inserting, or to interrupt such functions.").

Following *CollegeNet*, in *Z4 Techs., Inc. v. Microsoft Corp.*, 507 F.3d 1340 (Fed. Cir. 2007), the Federal Circuit again rejected arguments that "automatically" should exclude all user interaction, whether direct or indirect. In *Z4 Techs.* the Federal Circuit explained: "[N]othing in the claims or specification precludes user interaction in the *selection* or *initialization* of the automatic registration. Thus, the district court correctly rejected [Defendant's] attempt to exclude *any* user interaction from the claims." *Id.* at 1351; *see also Whitserve, LLC v. Computer Packages, Inc.*, 694 F.3d 10, 19–21 (Fed. Cir. 2012) (recognizing "[a] machine may still perform the claimed process automatically, even though a human might manually initiate or interrupt the process."). District courts have routinely followed the Federal Circuit's guidance with respect to the term "automatically," recognizing that indirect steps related to but not actually performing the limitation at issue may be done through human interaction. *See, e.g., Augme Techs., Inc. v. Pandora Media, Inc.*, No. 11–379–LPS, 2012 WL 6055010, at *7 (D. Del. Dec. 5, 2012) (Stark, J.) (relying on *CollegeNet* to reject the argument that the term "'automatically provided' necessarily excludes any human intervention"); *e-LYNXX Corp. v. Innerworkings, Inc.*, No. 1:10–CV–2535, 2012 WL 4484921, at *9–10 (M.D. Pa. Sept. 27, 2012) (finding that "automatically comparing" does not mean "without human intervention or input").

In sum, the intrinsic evidence, including the claims themselves, the specification, and the prosecution history, is consistent with established precedent and reflects that "automatically detect[ing]" precludes direct human interaction with respect to the detecting limitation itself, but does not preclude indirect human interaction related to the performance of the detection process, such as initiation or conclusion of the process by a user.

2. The Claims Are Not Limited to Detecting with an Intelligent Agent

The second issue with respect to the "automatically detect[ing]" limitation is Salesforce's attempt to add the requirement that detection may only occur "through the use of one or more intelligent agents." That requirement is not supported by the intrinsic evidence.

The pertinent limitation of claim 1 of the ’482 Patent recites: “a change management layer for automatically detecting changes that affect an application.” Ex. A, ’482 Patent, at 32:27–28. On its face, the limitation does not require intelligent agents. Instead, the claim explains that it is the “change management layer” that automatically detects changes, and says nothing about an additional requirement of an “intelligent agent.” Further, the claims evidence that the inventors were capable of claiming “intelligent agents” when they wanted to do so. Claims 8 and 28 of the ’482 Patent recite that “the change management layer further comprises one or more intelligent agents that detect changes that affect an application.” *Id.* at 32:59–61, 34:16–18. The fact that certain claims include an explicit requirement that the detecting be accomplished through “intelligent agents” evidences that not all claims are limited to “intelligent agents.” *See SRI Int’l v. Matsushita Elec. Corp. of Am.*, 775 F.2d 1107, 1122 (Fed. Cir. 1985) (en banc) (“It is settled law that when a patent claim does not contain a certain limitation and another claim does, that limitation cannot be read into the former claim in determining either validity or infringement.”); *Kara Tech.*, 582 F.3d at 1347 (“‘Differences among claims can … be a useful guide in understanding the meaning of particular claim terms.’ … [W]hen the inventor wanted to restrict the claims to require the use of a key, he did so explicitly.”).

The Federal Circuit has made clear that a requirement with no textual support in the claim language, such as requiring “intelligent agents,” can only be read into the claims in narrow and limited circumstances. As detailed above, a patentee must either set forth an express definition for a claim term or “disavow[] the full scope of a claim term either in the specification or during prosecution.” *Aventis*, 675 F.3d at 1330. Here, neither of those circumstances exist.

With respect to the specification, there can be no legitimate dispute that the specification includes no clear and express definition for “automatically detecting” that requires the use of intelligent agents. *See Bradium Techs. LLC v. Iancu*, 923 F.3d 1032, 1050 (Fed. Cir. 2019) (“[T]o act as its own lexicographer, a patentee must ‘clearly set forth a definition of the disputed claim term’ other than its plain and ordinary meaning.”). There is simply no definitional statement with respect to the phrase “automatically detecting” that “clearly express[es] an intent” that detection must only occur with an “intelligent agent.” *Id.*

Similarly, the specification does not contain an express disavowal of claim scope requiring the use of intelligent agents. Express disavowal requires the specification to "make[] clear" that the claim is limited to a particular form of the invention. *Hill-Rom Servs., Inc. v. Stryker Corp.*, 755 F.3d 1367, 1372 (Fed. Cir. 2014). Here, the specification does not remotely state that the claimed inventions require the use of intelligent agents. *See id.* (recognizing disavowal requires "clear" statements such as "the present invention requires …" or "the present invention is …" or "all embodiments of the present invention are …"). For example, the specification discusses "intelligent agents" as one of a variety of routines that can be "changed without requiring programming skill." Ex. A, '482 Patent, at 10:6–14[6] ("The Change Configuration functions support creation and change of End User functions through a variety of flexible and intelligent manual routines, such as ***intelligent agents, screens, fields, reports, documents and logic*** that can be changed without requiring programming skills."). The specification further explains that intelligent agents are embodiment dependent: "In one embodiment, the system is expressed as seven interacting menus … with the following functions[:] Intelligent Agent." *Id.* at 16:65–7:34. Far from using expressly limiting language, the specification uses permissive language in explaining that an "intelligent agent" "***can be used*** to identify changes in laws, statutes, ordinances, regulations and related issues." *Id.* at 20:3–4. The use of the permissive word "can" is the antithesis of an express statement that an intelligent agent is required.

Finally, consistent with the claim language and the ordinary rules of claim construction, the prosecution history evidences that the only claims limited to "intelligent agents" are those claims which contain limitations expressly directed to "intelligent agents." For example, in their appeal brief, the inventors argued that certain claims were separately patentable over Eager because Eager failed to disclose "intelligent agents." Ex. E, 08/28/07 Appeal Brief, at 21–25. Not surprisingly, the only claims the inventors distinguished from Eager based on the use of "intelligent agents" were those claims that expressly required "intelligent agents," *i.e.*, pending claims 11, 12 and 21. *Id.* at 30–31. The fact that the inventors distinguished Eager based on its lack of "intelligent agents" only

[6] In all quotations, emphasis is added unless otherwise indicated.

with respect to claims expressly limited to "intelligent agents" belies any suggestion that the inventors intended all claims to be limited to "intelligent agents." *See U.S. v. Telectronics, Inc.*, 857 F.2d 778, 782–83 (Fed. Cir. 1988) (arguments directed to claim that "explicitly contained such a limitation … cannot furnish a basis for restricting" a claim that does not contain such limitation).

3. The Term "Automatically Detecting" Is Not Indefinite

Salesforce's assertion that this term is indefinite lacks merit. AIT's construction is readily understandable and consistent with the intrinsic evidence, the claim language, and precedent. The fact that the claims are "readily understandable" to one of skill in the art is also evidenced by the expert declaration submitted by both parties' experts in this action. *See* Ex. F, Rosenberg CC Decl., ¶ 37; Ex. G, Bederson CC Decl., ¶ 77. Those declarations evidence a skilled artisan would be able to understand and apply the "automatically detecting" limitation without issue. *See Kinetic Concepts, Inc. v. Blue Sky Medical Group, Inc.*, 554 F. 3d 1010, 1022 (Fed. Cir. 2009) (finding claims sufficiently definite where accused infringer failed to establish "that a person of skill in the art would be unable to ascertain the meaning of the term.").

B. "changes that affect . . ."

Claim Term/Phrase	AIT Construction	Salesforce Construction
"changes that affect a particular application"/ "changes that affect an application" ('482 claims 1, 21)	No construction necessary – plain and ordinary meaning.[7]	"modifications to regulatory, technological, or social requirements stored in a third party repository that affect an application"
"changes that affect the information in the first portion of the server or the information in the second portion of the server" ('111 claim 13)	No construction necessary – plain and ordinary meaning.	"modifications to regulatory, technological, or social requirements stored in a third party repository that affect information about unique aspects of a particular application or functions common to various applications"

The dispute with respect to this term is straightforward: Salesforce wrongly seeks to add two separate requirements into the claims. First, Salesforce seeks to require that the claimed "changes" can only be changes to "regulatory, technological, or social requirements." Second, Salesforce seeks

[7] In an effort to streamline the issues, AIT no longer proposes a construction for these terms.

to add a requirement that the claimed "changes" must be stored in a "third party repository." Neither requirement is supported by the claim language itself, the specification, or the prosecution history.

The claim language does not support limiting the claims to the two requirements Salesforce seeks to add. Claim 1 of the '482 Patent recites "a change management layer for automatically detecting changes that affect an application." Ex. A, '482 Patent, at 32:27–28. The claim language does not state, suggest or imply that "changes that affect an application" can only be changes to regulatory, technology or social requirements. Similarly, the claim language makes no mention of a third party repository. As such, the two requirements Salesforce seeks to inject into the claims are not based on the ordinary meaning of the claim language.

The specification does not support limiting the claims to the two requirements Salesforce seeks to add. In order for any requirement to be read into the claims, Salesforce must point to either an expressly limiting definition or a clear and unmistakable surrender of claim scope. *Aventis*, 675 F.3d at 1330. Here, the specification plainly does not expressly require that changes may only relate to regulatory, technology or social requirements; or that these requirements be in a third party repository.

First, the specification does not unequivocally state that "changes" are limited to "regulatory, technological, or social requirements." To the contrary, the specification broadly describes the change management layer as "includ[ing] one or more change agents that … identify and bring to the user's attention relevant regulatory and non-regulatory changes … that may affect a user's business." Ex. A, '482 Patent, at 9:34–38. The term "changes" is used expansively:

> The invention provides an integrated system for managing data that is, or can be, constantly changing, because of ***changes*** in regulations, in the business environment, in technology and ***in any other factor that materially affects operations and/or information management requirements of a particular business***. …
>
> This invention monitors, responds to, and incorporates ***changes in, federal, state and local laws, statutes, ordinances and regulations*** (referred to collectively herein as "regulations") and ***changes in technology in one or more regulated areas of commercial activity***, such as environmental health and safety (EH&S), and food, drugs, cosmetics, medical devices and treatments ("FDCMTD").

Id. at 8:66–9:16. The specification thus recognizes that the changes may include "any … factor that materially affects operations and/or information management requirements of a particular business," and provides a wide range of examples. Nowhere does the specification indicate that the "changes" are limited by those representative examples.

Second, the specification does not contain a disclaimer requiring that all "changes" are detected from a "third party repository." To the contrary, the specification explains that the change layer may involve either "an intranet or the Internet." Ex. A, '482 Patent, at 16:18–19 ("[t]he change layer primarily involves ***an intranet or the Internet***"). An intranet is an internal network that is often walled off from third party repositories. Ex. H, Rosenberg Reply CC Decl., ¶ 33. Thus, the specification not only fails to unambiguously require that changes are limited to those detected from a "third party repository," it expressly teaches that changes can be detected from an intranet. Indeed, the specification explains that "[t]he internet is ***one source*** of information on regulatory change that is both prompt and cost-effective." Ex. A, '482 Patent, at 10:24–26. It further describes an exemplary process where an "Intelligent Agent launches one or more intelligent agents (IAs) to pursue ***internal*** and external Web activities." *Id.* at 19:66–67. The specification also recognizes that the invention can be used in "stand-alone or local applications," but it is not ***limited*** to such applications. Ex. A, '482 Patent at 22:29–31 ("Because the system is, or may be arranged to be, accessed and used through an Internet connection, the system is not limited to stand-alone or local applications."). When these statements are considered together, it is plain that the claimed "changes" are not limited to those detected from the Internet, let alone those detected from a "third party repository" not mentioned in the specification.[8]

C. "dynamically generate . . ."

Claim Term/Phrase	AIT Construction	Salesforce Construction
"dynamically generate a functionality and a user interface" ('111 claim 13)	"dynamically generate" means "generate or update when needed." No construction necessary	Indefinite, or in the alternative, requiring at least "generate [both a functionality and a user interface] immediately and concurrently without any

[8] Although not a preclusive finding, the PTAB recognized that the specification teaches an embodiment in which the detected changes are not limited to "external Web activities." Ex. I, PTAB Dec. 28, 2016 FWD, at 14 (relying on "intranet").

	for "a functionality and a user interface."	modification of software by a user"
"dynamically generated when the client computer connects to the server computer" / "dynamically re-generated" ('482 claims 1, 21)	"dynamically [re-]generated" means "generate or update when needed." No construction necessary for "when the client computer connects to the server computer."	Re-generated: "generated again after in [sic] initial generation" "When the client computer connects to the server computer" means "when a connection between a client computer and a server computer is initiated"

There are several disputes with respect to the "dynamically generate/re-generate" limitations. First, the parties dispute (1) what the phrase "dynamically generate" means. Second and third, Salesforce wrongly seeks to add requirements that dynamic generation must occur (2) "immediately and concurrently" and (3) "without any software modification by a user." Lastly, (4) Salesforce wrongly argues that the phrase "dynamically generate" is indefinite.

1. Dynamic Generation Occurs When Needed

With respect to the phrases "dynamically generate a functionality and a user interface" and "dynamically [re-]generate[d, ing]," "dynamically generate" means "generate or update when needed," and the rest of the phrases need no construction. This construction is consistent with the intrinsic evidence and the ordinary meaning of "dynamically."

With respect to the claim language, claim 1 of the '482 Patent recites "a user interface and functionality for the particular application is distributed to the browser application and dynamically generated when the client computer connects to the server computer." Ex. A, '482 Patent, at 32:31–34. Thus, the claim language on its face expressly specifies when the "dynamically generat[ing]" is performed, *i.e.*, "when the client computer connects to the server computer." Other independent claims recite similar limitations. *See id.* at 33:56–58 (claim 21); Ex. B, '111 Patent, at 33:29–34:4 (claim 13). Thus, every claim requiring dynamic generation also includes temporal limitations connected with such dynamic generation.

The phrases in the asserted claims which specify when dynamic generation is needed, *i.e.*, "when the client computer connects to the server computer" or when it "establishes a connection with the server computer," do not require further construction. The concepts of connecting and establishing a connection between two computers is sufficiently understandable to a jury such that no further construction is required. *See* Ex. F, Rosenberg CC Decl., ¶ 52.

The specification supports a conclusion that "dynamic" generation means generation "when needed." The specification teaches that the "Java data management layer … provides a graphical user interface for both the metadata layer and the business content layer." Ex. A, '482 Patent, at 15:5–7. The specification explains that "[t]he Java data management layer and thus what the end user sees … is generated ***as needed*** by a single program that interprets what a form will look like." *Id.* at 15:26–29. Consistently with this disclosure, the claims' use of the term "dynamically generat[ing]" refers to the ability of the inventive system to generate "a graphical user interface," *i.e.*, "what the end user sees," "as needed."

AIT's proposed construction is also consistent with the ordinary meaning of the words "dynamic" and "dynamically" and the meaning to one of skill in the art. In layperson's terms, something that is dynamic changes over time, in contrast to something that is static which tends not to change. In the software context, "dynamic code generation" refers to the generation of executable code at run time (*i.e.*, when it is needed) in contrast to static code which is generated once. *See* Ex. F, Rosenberg CC Decl., ¶ 48. Consistently, dictionaries have recognized the term dynamically may refer to changes that occur as needed. *See* Ex. J, The Microsoft Computer Dictionary, at 158 (recognizing dynamic when used "in describing both hardware and software … describes some action or event that occurs when and as needed").

2. Dynamic Generation Is Not Required to Occur Immediately and Concurrently

Salesforce's proposed requirement that "dynamically generating" must occur "immediately" and "concurrently" has no basis in the claim language or the specification and must be rejected.

First, Salesforce's construction is not supported by the claim language. There is no reason to conjure a requirement out of thin air regarding when dynamic generation must occur. As detailed above, the claims require that dynamic generation occurs either "when the client computer connects to the server computer" or when it "establishes a connection with the server computer." Ex. A, '482 Patent, at 32:33–34 (claim 1), 33:57–58 (claim 21); Ex. B, '111 Patent, at 34:3–4 (claim 13). The claims, on their face, lack any requirement that dynamic generation must occur "immediately" or "concurrently" with such connections.

Second, the specification fails to include any teaching requiring that dynamic regeneration may only occur "immediately" or "concurrently" with respect to any other process. Instead, as detailed above, consistent with the ordinary meaning of the term "dynamically," the specification states that the generation of the user interface may occur "as needed." Ex. A, '482 Patent, at 15:27–28. Far from containing an express disclaimer or lexicographical definition that would justify limiting the timing of dynamic generation such that it must occur "immediately" and "concurrently" with any other activities, the words "immediately" and "concurrently" are simply not used in the intrinsic evidence to describe, let alone limit, the timing of the claimed generation step. As it stands, it appears Salesforce seeks to create a claim requirement from whole cloth.

3. Dynamic Generation Does Not Preclude Modification by a User

The second limitation Salesforce seeks to graft onto the claims is a requirement that dynamic generation must occur "without any modification of software by a user." Salesforce's proposal is both inconsistent with the intrinsic record and misleading to the jury.

Salesforce does not specify what exactly it means by "without any modification of software by a user." To the extent Salesforce argues that an accused system must be programmed to infringe without ***requiring*** additional software programming, then there does not appear to be a dispute. Indeed, the Federal Circuit recognizes that infringement is avoided if an accused device is incapable of satisfying a claim limitation without reprogramming. *See Telemac Cellular Corp. v. Topp Telecom, Inc.*, 247 F.3d 1316, 1330 (Fed. Cir. 2001) ("[T]hat a device is capable of being modified to operate in an infringing manner is not sufficient, by itself, to support a finding of infringement."); *High Tech Med. Instrumentation, Inc. v. New Image Indus., Inc.*, 49 F.3d 1551, 1555–56 (Fed. Cir. 1995) (the fact that the accused device could be altered in a way that satisfies the claim term did not lead to infringement).

Consistent with settled precedent, the specification recognizes that the claimed invention must be capable of performing the claimed function "***without requiring*** the services of one or more programmers to re-program and/or recode the software items affected by the change." Ex. A, '482 Patent, at 8:41–43. Indeed, the specification teaches that although the system is capable of performing various processes without involving programmers, "small code segments" may be used to perform such processes. *Id.* at 12:50–52 ("This approach is implemented using intuitive, user-friendly, dialog-based screens and using small code segments to define business logic."). The prosecution history reiterates that the inventive system is ***capable*** of performing various functions, including generating an application "***without requiring*** (re)programming of underlying software." Ex. E, 08/28/07 Appeal Brief, at 3.

Salesforce's proposed construction departs in one important way from the law of infringement and the intrinsic evidence: The law of infringement and the intrinsic evidence recognized that reprogramming cannot be "required," but it is permitted nevertheless. Indeed, the specification and the prosecution history both use the phrase "without requiring," yet that phrase is absent from Salesforce's construction. As such, Salesforce's construction appears to exclude not only any system that requires reprogramming to perform the dynamic generation process, but also any system that permits, but does not require, such reprogramming. Any such argument would be inconsistent with the teachings of the Asserted Patents.

4. The Term "Dynamically Generat[ing]" Is Not Indefinite

Despite proposing a construction for this term, Salesforce urges that the term "dynamically generate" is not amenable to construction. The evidence belies Salesforce's argument. Both parties' experts have submitted declarations evidencing that the term "dynamically generat[ing]" is understandable to one of skill in the art. Ex. F, Rosenberg CC Decl., ¶¶ 46–48; Ex. G, Bederson CC Decl., ¶¶ 122–128; Ex. H, Rosenberg Reply CC Decl., ¶ 45–47. Salesforce's expert has merely opined that the term is indefinite if the Court rejects Salesforce's attempts to read in various requirements. *See* Ex. G, Bederson CC Decl., ¶¶ 127–128. However, as a legal matter, claims are not indefinite merely because they fail to contain requirements an accused infringer would like. *See Ultimax Cement Mfg. Corp. v. CTS Cement Mfg. Corp.*, 587 F.3d 1339, 1352 (Fed. Cir. 2009) ("Merely claiming broadly does not render a claim insolubly ambiguous, nor does it prevent the public from understanding the scope of the patent.").

D. "layer" / "portion of the server" or "portion"

Claim Term/Phrase	AIT Construction	Salesforce Construction
"layer"	"a set of functionally or	Indefinite, or in the alternative,

('482 claims 1, 3, 5, 10, 20, 21, 23, 25, 30, 40)	logically related software components"	requiring at least "a group of data and/or functions that is separate and distinct from other such groups"
"portion of the server" / "portion" ('111 claims 13–17)	"a functionally or logically related subset of one or more server computers"	Indefinite, or in the alternative, requiring at least "a subset of one server computer separate and distinct from other subsets"

AIT's proposed construction for the term "layer" is "a set of functionally or logically related software components." That construction is consistent with the ordinary meaning of the word "layer" in the art of the '482 Patent and its usage in the specification. Similarly, AIT's proposed construction for the related term "portion [of the server]" as "a functionally or logically related subset of one or more server computers" is consistent with the intrinsic evidence. Salesforce seeks to inject into these terms two extraneous requirements: (i) that "layers" or "portions" be "separate and distinct" from each other; and (ii) that a "portion" be "a subset of ***one*** server computer." Salesforce further argues indefiniteness, which is nonsensical in view of the testimony of both sides' experts.

The meaning of "layer" is readily apparent from the specification of the '482 Patent. The patent describes a system with a layered architecture, including a "change management layer," a "Java data management layer," a "metadata layer," and a "business content layer." Ex. A, '482 Patent, at 9:33–48. The structure and function of each of these "layers" is described in detail in the specification. *Id.* at 12:15–16:60. Fig. 1 of the '482 Patent shows diagrammatically how these different layers logically relate to each other. *Id.* at Fig. 1.

The '482 Patent does not provide an express definition of "layer," however the meaning of this term is well-known and well-defined in the software field. *See* Ex. F, Rosenberg CC Decl., ¶ 35. Consistent with its use in the specification, a layer is generally understood to be a logical structuring mechanism for the elements that make up a software system. *Id.* Thus, one of ordinary skill in the art would understand that "layer" is used in the '482 Patent to refer to "a set of functionally or logically related software components." *Id.* As detailed above, this concept of layers being comprised of functionally or logically related items is consistent with the specification.

The related term "portion" should be construed consistently with "layer." The ordinary meaning of "portion" is not in dispute: it is "[a] section or quantity within a larger thing; a part of a whole." Ex. K, American Heritage Dictionary of the English Language (4th ed. 2000), at 1368. Consistently with the plain meaning of the word "portion," in the context of the '111 Patent a "portion [of a server]" is "a functionally or logically related subset of one or more server computers." Indeed, Fig. 1 discloses logical or functional portions of a server and not physically separate portions. Ex. A, '482 Patent, at Fig. 1.

Layers and portions are not required to be physically separate and distinct. Salesforce argues that a "layer" must be "separate and distinct from other [layers]," *i.e.*, precluding any overlap between "layers." This argument simply ignores that the claimed "layers" are software components and not physical articles. Indeed, the specification does not require that "layers" be physically separate and distinct from each other. To the contrary, as detailed above, the Fig. 1 embodiment shows a logical, not physical, relationship between layers and demonstrates that the layers are all software components that functionally work together. Further, the specification recognizes that layers are not required to be physically separate and distinct from each other, expressly teaching that two layers may "together serve as a standard interface system." Ex. A, '482 Patent, at 16:4–5. The specification further explains that "[t]he relationship between the business content layer and the Java data management layer may be characterized as a 'data mapping'" (*id.* at 15:19–21), which requires information in each layer to be associated with each other.

Salesforce's position with respect to the related term "portion [of the server]" similarly lacks merit. Nothing in the plain meaning of the word "portion" requires it to be "separate and distinct from other [portions]," *i.e.*, non-overlapping. Nor do the claims, specification, or prosecution history suggest that "portions" must be completely physically separate or distinct.

A portion does not have to be a subset of a single server. Salesforce's construction is also wrong because it requires that a "portion" be "a subset of ***one*** server computer," rather than "***one or more*** server computers" as proposed by AIT. Claim 13 recites "***a server*** accessible by a browser executed on a client device, the server including" four "portions." Ex. B, '111 Patent, at 33:20–22. In patent parlance, the phrase "a server" means ***one or more*** servers. *Baldwin Graphic Systems, Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1342 (Fed. Cir. 2008) ("That 'a' or 'an' can mean 'one or more' is best described as a rule, rather than merely as a presumption or even a convention."). Salesforce's proposed construction flies in the face of this rule of patent law.

"Layer" and "portion" are not indefinite. The scope of these terms is reasonably certain to those of ordinary skill in the art. *See Nautilus*, 572 U.S. at 910. Both sides' experts agree that "layer" is a term of art in the field of the '482 Patent. *See* Ex. F, Rosenberg CC Decl., ¶ 35; Ex. G, Bederson CC Decl., ¶ 130. Salesforce's expert merely opines that if the term is construed as proposed by AIT (*i.e.*, if the term is susceptible to AIT's construction) then the term is not susceptible to construction. Ex. G, Bederson CC Decl., ¶ 141. On its face, Salesforce's position is nonsensical. If the Court adopts AIT's construction of "layer" and "portion" then, by definition, those terms are subject to construction. *See Ultimax Cement Mfg. Corp.*, 587 F.3d at 1352 ("claiming broadly does not render a claim insolubly ambiguous ….").

E. "unique aspects of a particular application" / "functions common to various applications"

Claim Term/Phrase	AIT Construction	Salesforce Construction
"unique aspects" ('482 claims 1, 21) ('111 claim 13)	No construction necessary – plain and ordinary meaning.[9]	Indefinite
"information about user interface elements and one or more functions common to various applications" / "information about the user interface and functions common to a variety of applications" ('111 claim 13) ('482 claims 1, 21)	No construction necessary – plain and ordinary meaning.	Indefinite, or in the alternative, requiring at least "information about user interface components and functions used by multiple different applications, excluding any unique aspects of those applications"

The primary dispute with respect to these terms centers on Salesforce's contention that the terms "unique" and "common" are not amenable for construction. Yet, in the context of the claims, the terms "unique" and "common" are readily understandable by both those of skill in the art and lay readers, and there is no need to further explain the meaning of these terms. Further, Salesforce's construction for the "common" terms as "excluding any unique aspects" makes little sense as the claim already requires "common" information to be "common to a variety of applications."

[9] In an effort to streamline the issues, AIT no longer proposes a construction for these terms.

Salesforce's indefiniteness argument can be resolved based on the claim language itself. For example, claim 1 of the '482 Patent recites "a first layer . . . containing information about the unique aspects of a particular application" and "a second layer . . . containing information about the user interface and functions common to a variety of applications." Ex. A, '482 Patent, at 32:15–20; *see also id.* at 33:39–43; Ex. B, '111 Patent at 33:23–28. Thus, in the context of the claim, "unique aspects of a particular application" is contrasted with aspects "common to a variety of applications." Based on the context of the claim limitations, "unique aspects" are simply aspects of an application that are not "common" to multiple applications. Salesforce's indefiniteness argument simply ignores that the words "unique" and "common" must be considered in the context of how they are used in the claim. *ACTV, Inc. v. Walt Disney Co.*, 346 F.3d 1082, 1088, 1090 (Fed. Cir. 2003) ("[T]he context of the surrounding words of the claim also must be considered in determining the ordinary and customary meaning" of terms in a claim); *IGT v. Bally Gaming Int'l, Inc.*, 659 F.3d 1109, 1117 (Fed. Cir. 2011) ("Extracting a single word from a claim divorced from the surrounding limitations can lead construction astray.").

Even if the use of the word "unique" was not clear from its context in the claims, the mere fact that the word "unique" may be considered a relative term does not mean that the claim is indefinite. *See, e.g., Exmark Mfg. Co. Inc. v. Briggs & Stratton Power Prod. Grp., LLC*, 879 F.3d 1332, 1346 (Fed. Cir. 2018) ("All that is required is some standard for measuring the term of degree."). Indeed, courts have no difficulty construing the term "unique" even in the absence of clarifying language in the claims. *See, e.g., Intellectual Ventures I, LLC v. Motorola Mobility LLC*, 81 F. Supp. 3d 356, 369–70 (D. Del. 2015) (claim term "unique transaction identifier" was not indefinite); *In re Maxim Integrated Products, Inc.*, No. 12-244, 2014 WL 3696137, at *11 (W.D. Penn. July 23, 2014) (claim term "substantially unique electronically readable identification number" not indefinite).

The specification is consistent with the claimed recitation of "unique" and "common" aspects of an application. For example, the Asserted Patents describe a "business content layer" that contains information "***specific to the particular business operations*** of interest to the user." Ex. A, '482 Patent, at 9:46–48. The specification also discloses a "metadata layer" that "provides and/or defines data about ***every feature of the user interface*** including, without limitation, tools, worklists, data entry forms, reports, documents, processes, formulas, images, tables, views, columns, and other structures and functions." *Id.* at 9:41–46. The specification further explains that "the metadata architecture … stores ***all of the information used to create the front-end business application*** and manage the back-end business database." *Id.* at 10:15–17. Thus, consistent with the claims, the specification teaches storing features of a "particular application," *e.g.*, those that are unique to that application, and features used to "create the front-end business applications," *e.g.*, those that are common to multiple applications. *See* Ex. F, Rosenberg CC Decl., ¶ 50.

One of ordinary skill in the art reading the claims would understand the boundary of the terms "unique" and "common" and be able to apply them with reasonable certainty. As explained by Mr. Rosenberg, when the terms "unique" and "common" are considered in light of the claim language they are readily understandable. Ex. H, Rosenberg Reply CC Decl., ¶ 51. Consistent with Dr. Rosenberg's testimony and the language of the claims, during the IPR proceedings, expert declarations were submitted by Salesforce and RPX evidencing that the meaning of "unique" and "common" in the context of the claims would be readily ascertainable to one of ordinary skill in the art. *See, e.g.*, Ex. M, Crovella PTAB Decl., ¶¶ 31–33.

Lastly, with respect to the "common" terms, Salesforce separately proposes adding a requirement that aspects "common to various applications" do not include "unique aspects" of those applications. That requirement appears to serve no purpose. To the extent information is common to various applications, by definition, such information is not unique to a particular application.

F. "business content database"

Claim Term/Phrase	AIT Construction	Salesforce Construction
"business content database" ('482 claims 3, 23)	"a data store containing data specific to particular business operations"	Indefinite

As recognized by AIT's construction, a "business content database" is "a data store containing data specific to particular business operations." Salesforce contends that the straightforward term "business content database" is not susceptible to construction.

Claims 3 and 23 of the '482 Patent recite "a business content database having data about one or more different predetermined business applications." Ex. A, '482 Patent, at 32:41–43, 33:66–67. On its face, a "business content database" is a database containing data relating to business content. The specification explains that business content includes "business knowledge, logical designs, physical designs, physical structures, relationships, and data associated with a selected area of business activity." *Id.* at 12:16–19. In other words, as the claim term implies, the specification teaches the "business content database" is a repository that contains various types of data "associated with a selected area of business activity." *Id.* That is consistent with AIT's proposed construction of "a data store containing data specific to particular business operations."

The ordinary meaning of "business content database" is readily understandable to a skilled person as well as to a lay juror. First, there can be no dispute concerning the meaning of "database." *See* Ex. A, '482 Patent at 29:50–52 ("A 'database' is a collection or group of objects that holds various related information items."). Adding the modifier "business content" to the word "database" does not render this phrase indefinite. Instead, "business content" is simply an adjective used to describe the type of information stored in the claimed database. The use of adjectives in this context is hardly unique and does not somehow render the claimed phrase unclear to a skilled person. *See* *Digeo, Inc. v. Audible, Inc.*, No. 05-CV-464-JLR, 2006 WL 828861, at *13 (W.D. Wash. Mar. 27, 2006) (recognizing a "transactional database" has an ordinary meaning that is "a structured computer memory for storing and accessing data related to transactions"); *Soverain Software LLC v. Amazon.com, Inc.*, No. 04-CV-14-LD, 2005 WL 6225276, at *10 (E.D. Tex. Apr. 7, 2005) ("'merchant database' means 'a database of or related to a merchant'").

G. "logical design" / "physical design" / "physical structure"

Claim Term/Phrase	AIT Construction	Salesforce Construction
"logical design" ('482 claim 24) ('111 claim 15)	No construction necessary – plain and ordinary meaning.[10]	"an arrangement of data in a series of logical relationships referred to as entities or attributes"
"physical design" ('482 claim 24) ('111 claim 15)	No construction necessary – plain and ordinary meaning.	"description of a physical database including tables and constraints"
"physical structure"	No construction necessary –	"structure of a database that can be seen and operated on by the operating

[10] In an effort to streamline the issues, AIT no longer proposes a construction for these terms.

('482 claim 24) ('111 claim 15)	plain and ordinary meaning.	system, such as the physical files stored on a disk"

The plain language of the claims is dispositive as to the meaning of these terms. For example, claim 23 of the '482 Patent recites "a business content database having data about one or more different predetermined business applications." Ex. A, '482 Patent, at 33:65–67. Claim 24 further requires that "the data [of claim 23] … comprises one or more of business knowledge, ***logical designs, physical designs, physical structures*** and relationships associated with the predetermined business application." *Id.* at 34:1–5. As such, "logical designs, physical designs, [and] physical structures" are types of "data about one or more different predetermined business applications" within the "business content database." No further construction is necessary.

Salesforce's constructions explode these six words into a total of 49 words plucked from an Oracle software manual (not a dictionary or treatise) that is not referenced in the specification. *See* Ex. N at 2-2, 3-2. Based on that manual alone, Salesforce seeks to limit the claim scope to an arrangement of data, or a ***description/structure*** of a database. Such usage is contrary to the claim language, which specifies the logical designs, physical designs, and physical structures are types of "data" that may be contained in the "business content database." As such, the Court should reject Salesforce's proposal as improperly narrow and confusing.

H. "builder module"

Claim Term/Phrase	AIT Construction	Salesforce Construction
"builder module" ('482 claim 10)	"a software tool to construct an application or part of an application"[11]	"self-contained unit of software capable of generating part of an application"

The parties dispute whether the term "builder module" is a "software tool," as AIT proposes, or whether the "builder module" must be a "self-contained unit of software," as Salesforce argues. The term "builder module" appears in claim 10 of the '482 Patent. That claim specifies that the "server" comprises, among other things, "a builder module for permitting a user to build a user interface for a particular application using the second layer." Ex. A, '482 Patent, at 33:1–4. Thus,

[11] Previously, AIT had asserted that the application had to be constructed "from metadata." In an effort to narrow the parties' dispute, AIT is no longer pressing that position.

based on the plain language of the claim, the builder module must be software that is used as a tool, *i.e.*, "a software tool," to construct, *i.e.*, "build," the "user interface … using the second layer." Unasserted claims 11–18 of the '482 Patent similarly specify that the software required by the "builder module" is a tool to construct or build various objects such as forms, using a "form builder," reports using a "report builder," and documents using a "document builder." *Id.* at 33:5–30. As such, consistent with the claims of the Asserted Patents, the "builder module" should be construed as "a software tool to construct an application or part of an application."

AIT's construction is consistent with the specification and its discussion of the software tools used to build an application. The specification explains:

> Within the Java management layer, ***configuration tools*** take the place of a programmer and ***define various end user functions*** in terms of metadata, and metadata definitions are used to implement the desired end user functions. Within the metadata layer, the relevant items (data entry forms, etc.) in the business content layer are defined, regulatory and non-regulatory changes in these items are implemented, and access thereto is provided.

Id. at 9:49–56; *see also id.* at 16:5–12; 16:35–47 and Fig. 6 (showing procedure for creating a data entry form); *id.* at 20:63–21:58 (detailing configuration tools). As such, the term "builder module" is disclosed in the specification in the same manner it is used in the claims to refer to "a software tool" that is used to construct an application or part of an application. *See* Ex. F, Rosenberg Decl., ¶ 88.

Salesforce's proposal that a "builder module" is a "self-contained unit of software" is unsupported by the specification and confusing. There is nothing in the specification that requires a "builder module" to be a "self-contained unit of software." As it stands, it appears that Salesforce simply created a requirement from whole cloth.

V. CONCLUSION

For the foregoing reasons, AIT respectfully requests the Court to adopt its proposed constructions of the abovementioned disputed terms.

Dated: April 6, 2021

/s/ Andrea Pacelli

Michael A. Burke, Esq.
ROBISON, BELAUSTEGUI, SHARP & LOWA
Professional Corporation (Resident Counsel)
71 Washington Street
Reno, Nevada 89503

Andrea Pacelli, Esq. (*pro hac vice*)
Mark S. Raskin, Esq. (*pro hac vice*)
Michael DeVincenzo, Esq. (*pro hac vice*)
Elizabeth Long, Esq. (*pro hac vice*)
Eric Berger, Esq. (*pro hac vice*)
KING & WOOD MALLESONS LLP
500 Fifth Ave., 50th Floor
New York, New York 10110
Telephone: (212) 319-4755
Email: andrea.pacelli@us.kwm.com
mark.raskin@us.kwm.com
michael.devincenzo@us.kwm.com
elizabeth.long@us.kwm.com
eric.berger@us.kwm.com

Steven C. Sereboff, Esq. (*pro hac vice*)
SoCAL IP LAW GROUP LLP
1332 Anacapa, Suite 201
Santa Barbara, CA 93101
Telephone: (805) 230-1356
Email: ssereboff@socalip.com

Attorneys for Plaintiff
Applications in Internet Time, LLC

CERTIFICATE OF SERVICE

The undersigned certifies that on April 6, 2021 I caused the foregoing **PLAINTIFF APPLICATIONS IN INTERNET TIME, LLC'S OPENING CLAIM CONSTRUCTION BRIEF** to be filed with the Clerk of the Court by using the CM/ECF system which will send a Notice of Electronic Filing to the following counsel of record:

Leigh T. Goddard
lgoddard@mcdonaldcarano.com

John J. Frankovich
jfrankovich@mcdonaldcarano.com

Kevin P.B. Johnson
kevinjohnson@quinnemanuel.com

Ray R. Zado
rayzado@quinnemanuel.com

Sam S. Stake
samstake@quinnemanuel.com

Attorneys for Defendant salesforce.com, Inc.

Dated: April 6, 2021

/s/ Andrea Pacelli
Andrea Pacelli